Text to brain: predicting the spatial distribution of neuroimaging observations from text reports

Jérôme Dockès[1], Demian Wassermann[1], Russell Poldrack[2], Fabian Suchanek[3], Bertrand Thirion[1], and Gaël Varoquaux[1]

[1]INRIA, CEA, Université Paris-Saclay, [2]Stanford University, [3]Télécom ParisTech

Abstract. Despite the digital nature of magnetic resonance imaging, the resulting observations are most frequently reported and stored in text documents. There is a trove of information untapped in medical health records, case reports, and medical publications. In this paper, we propose to mine brain medical publications to learn the spatial distribution associated with anatomical terms. The problem is formulated in terms of minimization of a risk on distributions which leads to a least-deviation cost function. An efficient algorithm in the dual then learns the mapping from documents to brain structures. Empirical results using coordinates extracted from the brain-imaging literature show that i) models must adapt to semantic variation in the terms used to describe a given anatomical structure, ii) voxel-wise parameterization leads to higher likelihood of locations reported in unseen documents, iii) least-deviation cost outperforms least-square. As a proof of concept for our method, we use our model of spatial distributions to predict the distribution of specific neurological conditions from text-only reports.

1 Introduction

Hundreds of thousands of studies, case reports, or patient records, capture observations in human neuroscience, basic or clinical. Statistical analysis of this large amount of data could provide new insights. Unfortunately, most of the spatial information that these data contain is difficult to extract *automatically*, because it is hidden in unstructured text, in sentences such as: "[...] in the anterolateral temporal cortex, especially the temporal pole and inferior and middle temporal gyri" [1].

This data cannot be processed easily by a machine, as a machine does not know where the temporal cortex is. As we will show, simply looking up such terms in atlases does not suffice. Indeed, even atlases disagree [2]. Furthermore, joint processing of many reports faces varying terminologies, with regions represented in different atlases that differ and overlap. Finally, not all terms in a report carry the same importance, and practitioners use terms that are not the exact labels of any atlas. Coordinate-based meta-analyses capture the spatial distribution of a term from the literature [3,4], but they also lack a model to combine terms.

Here, we propose to map case reports automatically to the brain locations that they discuss: we learn mappings of anatomical terms to brain regions from medical publications. We propose a new learning framework for translating anatomical terms to brain images – a process that we call “encoding”. We learn such a mapping, quantify its performance, and compare possible choices of representation of spatial data. We then show in a proof of concept that our model can predict the brain area for textual case reports.

2 Methods: formalizing text-to-brain-map translation

2.1 Problem setting: from text to spatial distributions

We want to predict the likelihood of the location of relevant brain structures described in a document. For this purpose, we perform supervised learning on a corpus of brain-imaging studies, each containing: (i) a text, and (ii) the locations – *i.e.* the stereotactic coordinates – of its observations. Indeed, Functional Magnetic Resonance Imaging (fMRI) studies report the coordinates of activation peaks (*e.g.,* [5, Table 1]), and Voxel Based Morphometry (VBM) analyses report the location of differences in gray matter density (*e.g.,* [1, Table 2]). Following neuroimaging meta-analyses [3], we frame the problem in terms of spatial distributions of observations in the brain. In a document, observed locations $\mathcal{L} = \{l_a \in \mathbb{R}^3, a = 1 \dots c\}$ are sampled from a probability density function (pdf) p over the brain. **Our goal is to predict this pdf p from the text $\mathcal{T}$.** We denote q our predicted pdf. A predicted pdf q should be close to p, or take high values at the coordinates actually reported in the study: $\prod_{l \in \mathcal{L}} q(l)$ must be large. In a supervised learning setting, we start from a collection of studies $\mathcal{S} = (\mathcal{T}, \mathcal{L})$, with $\mathcal{T}$ the text and $\mathcal{L}$ the locations. Building the prediction engine then entails the choice of a model relating the predicted pdf p to the text $\mathcal{T}$, the choice of a loss, or data-fit term, and some regularization on the model parameters. We now detail how we make each of these choices to construct a prediction.

Model. We start by modelling the dependency of our spatial pdf q on the study text $\mathcal{T}$. This entails both choosing a representation for q and writing it as a function of the text. While q is defined on a subvolume of $\mathbb{R}^3$, the brain volume, we build it using a partition to work on a finite probably space: this can be either a regular grid of voxels or a set of anatomical regions (i.e. an atlas) $\mathcal{R} = \{\mathcal{R}_k, k = 0 \dots m\}$. As such a partitioning imposes on each region to be homogeneous, q is then formally written on $\mathbb{R}^3$ in terms of the indicator functions of the parts[1]: $\{r_k = \frac{\mathbb{I}_k}{\|\mathbb{I}_k\|_1},\ k = 1 \dots m\}$. Importantly, the volume of each part $\|\mathbb{I}_k\|_1$ appears as a normalization constant.

To link q to the text $\mathcal{T}$ of the study, we start by building a term-frequency vector representation of $\mathcal{T}$, which we denote $\boldsymbol{x} \in \mathbb{R}^d$. d is the size of our vocabulary of English words $\mathcal{W} = \{w_t\}$, and $\boldsymbol{x}_t$ is the frequency of word w_t in the text. We assign to each atlas region a weight that depends linearly on $\boldsymbol{x}$:

$$q(z) = \sum_{t=1}^{d} \sum_{k=1}^{m} \boldsymbol{x}_t \boldsymbol{\beta}_{t,k} r_k(z) \quad \forall z \in \mathbb{R}^3 \tag{1}$$

[1] $\mathcal{R}_0$ denotes the volume outside of the brain, or background, on which q is 0.

where $\boldsymbol{\beta} \in \mathbb{R}^{d \times m}$ are model parameters, which we will learn.

Using an atlas is a form of regularization: constraining the prediction to be in the span of $\{r_k\}$ reduces the size of the search space. Fine partitions, *e.g.* atlases with many regions or voxel grids, yield models with more expressive power, but more likely to overfit. Choosing an atlas thus amounts to a bias-variance tradeoff.

Label-constrained encoder. A simple heuristic to turn a text into a brain map is to use atlas labels and ignore interactions between terms. The probability of a region is taken to be proportional to the frequency of its label in the text. The vocabulary is then the set of labels: $d = m$. As the word w_k is the label of $\mathcal{R}_k$, $\boldsymbol{\beta}$ is diagonal. For example, for a region $\mathcal{R}_k$ in the atlas labelled "parietal lobe", the probability on $\mathcal{R}_k$ depends only on the frequency of the phrase "parietal lobe" in the text. We call this model *label-constrained encoder*.

2.2 Loss function: measuring errors on spatial distributions

Strategy. We will fit the coefficients $\boldsymbol{\beta}$ of our model, see Eq. (1), by minimizing a risk $\mathcal{E}(p, q)$: the expectation of a distance between p and q.

A plugin estimator of p. We do not have access to the true pdf, p; we need a plugin estimator, which we denote $\hat{p}$. By construction of our prediction q, the best approximation of p we can hope for belongs to the span of our regions $\{r_k\}$. Hence, we build our estimator $\hat{p}$ in this space, setting the probability of a region to be proportional to the number of coordinates that fell inside it:

$$\hat{p} = \sum_{k=1}^{m} \frac{|\{a, \mathbb{I}_k(l_a) = 1\}|}{c} r_k \;=\; \sum_{k=1}^{m} \frac{1}{c} \sum_{a=1}^{c} \mathbb{I}_k(l_a) r_k \;\triangleq\; \sum_{k=1}^{m} \hat{\boldsymbol{y}}_k r_k \quad . \tag{2}$$

When regions are voxels, there are too many regions and too few coordinates. Hence we use Gaussian Kernel Density Estimation (KDE) to smooth the estimated pdf[2]. Our supplementary material details a fast KDE implementation.

Choice of $\mathcal{E}$. We use two common distance functions for our loss. The first is Total Variation (TV), a common distance for distributions. Note that p defines a probability measure on the finite sample space $\mathcal{R}$, $\mathcal{P}(\mathcal{R}_k) = \int_{\mathcal{R}_k} p(z) dz$, where $\mathcal{R} = \{\mathcal{R}_k, k = 1 \dots m\}$ and $\mathcal{R}_k = \operatorname{supp}(r_k)$. q defines Q in the same way. Then,

$$\mathrm{TV}(\mathcal{P}, \mathcal{Q}) = \sup_{\mathcal{A} \subset \mathcal{R}} |\mathcal{P}(\mathcal{A}) - \mathcal{Q}(\mathcal{A})| \quad . \tag{3}$$

Since $\mathcal{R}$ is finite, a classical result (see [6]) shows that this supremum is attained by taking $\mathcal{A} = \{\mathcal{R}_k | \mathcal{P}(\mathcal{R}_k) > \mathcal{Q}(\mathcal{R}_k)\}$ (or its complementary) and:

$$\mathrm{TV}(\mathcal{P}, \mathcal{Q}) = \frac{1}{2} \sum_{k=1}^{m} |\mathcal{P}(\mathcal{R}_k) - \mathcal{Q}(\mathcal{R}_k)| = \frac{1}{2} \int_{\mathbb{R}^3} |p(z) - q(z)| dz \quad . \tag{4}$$

[2] Using an atlas is also a form of KDE, with kernel $(z, z') \mapsto 1/\|\mathbb{I}_k\|_1$ if z and z' belong to the same region $\mathcal{R}_k, k \in \{1, \dots m\}$, 0 otherwise.

The TV is half of the ℓ_1 distance between the pdfs. $\|\hat{p}-q\|_1$ is therefore a natural choice for our loss. The second choice is $\|\hat{p}-q\|_2^2$, which is a popular distance and has the appeal of being differentiable everywhere.

Factorizing the loss. Let us call v_k the volume of r_k, i.e. the size of its support: $v_k \triangleq \|\mathbb{I}_k\|_1,\ k = 1 \dots m$. Remember that $r_k = \frac{1}{v_k}\mathbb{I}_k$. Our loss can now be factorized (see supplementary material for details):

$$\int_{\mathbb{R}^3} \delta(\hat{p}(z) - q(z))dz = \sum_{k=1}^{m} v_k \delta \left(\frac{\hat{\boldsymbol{y}}_k}{v_k} - \frac{\sum_{t=1}^{d} \boldsymbol{x}_t \boldsymbol{\beta}_{t,k}}{v_k} \right) \tag{5}$$

Here, δ is either the absolute value of the difference or the squared difference.

2.3 Training the model: efficient minimization approaches

To set the model parameters $\boldsymbol{\beta}$, we used n example studies $\{\mathcal{S}_i = (\mathcal{T}_i, \mathcal{L}_i),\ i = 1 \dots n\}$. We learn $\boldsymbol{\beta}$ by minimizing the empirical risk on $\{S_i\}$ and an ℓ_2 penalty on $\boldsymbol{\beta}$. We add to the previous notations the index i of each example: p_i, q_i, $\hat{\boldsymbol{y}}_i$, $\boldsymbol{x}_i$. $\hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times m}$ is the matrix such that $\hat{\boldsymbol{Y}}_i = \hat{\boldsymbol{y}}_i$, and $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ such that $\boldsymbol{X}_i = \boldsymbol{x}_i$.

Case $\delta = \ell_2^2$. The empirical risk is

$$\sum_{i=1}^{n} \sum_{k=1}^{m} \left(\frac{\hat{\boldsymbol{Y}}_{i,k}}{\sqrt{v_k}} - \sum_{t=1}^{d} \frac{1}{\sqrt{v_k}} \boldsymbol{X}_{i,t} \boldsymbol{\beta}_{t,k} \right)^2 . \tag{6}$$

Defining $\boldsymbol{Y'}_{:,k} = \frac{\hat{\boldsymbol{Y}}_{:,k}}{\sqrt{(v_k)}}$ and $\boldsymbol{\beta'}_{:,k} = \frac{\boldsymbol{\beta}_{:,k}}{\sqrt{(v_k)}}$, with an ℓ_2 penalty, the problem is:

$$\underset{\boldsymbol{\beta'}}{\operatorname{argmin}} \left(\|\boldsymbol{Y'} - \boldsymbol{\beta'}\boldsymbol{X}\|_2^2 + \lambda \|\boldsymbol{\beta'}\|_2^2 \right) \tag{7}$$

where $\lambda \in \mathbb{R}_+$. This is the least-squares ridge regression predicting $\hat{p}$ expressed in the orthonormal basis of our search space $\{\frac{r_k}{\|r_k\|_2}\}$.

Case $\delta = \ell_1$. The empirical risk becomes

$$\sum_{i=1}^{n} \sum_{k=1}^{m} |\hat{\boldsymbol{Y}}_{i,k} - \sum_{t=1}^{d} \boldsymbol{X}_{i,t} \boldsymbol{\beta}_{t,k}| \tag{8}$$

This problem is also known as a least-deviations regression, a particular case of quantile regression [7], [8]. Unlike ℓ_2 regression, which provides an estimate of the conditional mean of the target variable, ℓ_1 provides an estimate of the median. Quantile regression has been studied (e.g. by economists), as it is more robust to outliers and better-suited than least-squares when the noise is not normally distributed [7]. Adding an ℓ_2 penalty, we have the minimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\hat{\boldsymbol{Y}} - \boldsymbol{X}\boldsymbol{\beta}\|_1 + \lambda \|\boldsymbol{\beta}\|_2^2 \right) \tag{9}$$

Unpenalized quantile regression is often written as a linear program and solved with the simplex algorithm [9], iteratively reweighted least squares, or interior point methods [10]. [11] uses a coordinate-descent to solve a differentiable approximation of the quantile loss (the Huber loss) with elastic-net penalty. Here, we minimize Eq. (9) via its dual formulation (c.f. supplementary material):

$$\hat{\boldsymbol{\nu}} = \underset{\boldsymbol{\nu}}{\operatorname{argmax}} \left(Tr(\boldsymbol{\nu}^T \hat{\boldsymbol{Y}} - \frac{1}{4\lambda} \boldsymbol{\nu}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{\nu}) \right) \quad \text{s.t. } \|\boldsymbol{\nu}\|_\infty \leq 1, \tag{10}$$

where $\boldsymbol{\nu} \in \mathbb{R}^{n \times m}$. The primal solution is given by $\hat{\boldsymbol{\beta}} = \frac{\boldsymbol{X}^T \hat{\boldsymbol{\nu}}}{2\lambda}$. As the dual loss g is differentiable and the constraints are *bound* constraints, we can use an efficient quasi-Newton method (L-BFGS, [12]). g and its gradient are fast to compute as $\boldsymbol{X}$ is sparse. λ is set by cross-validation on the training set. We use warm-start on the regularization path (decreasing values for λ) to initialize each problem.

Training the label-constrained encoder. The columns of $\boldsymbol{\beta}$ can be fitted independently from each other. If we want $\boldsymbol{\beta}$ to be diagonal, we only include one feature in each regression: we fit m univariate regressions $\hat{\boldsymbol{y}}_{:,k} \simeq \boldsymbol{X}_{:,k} \boldsymbol{\beta}_{k,k}$.

2.4 Evaluation: a natural model-comparison metric

Our metric is the mean log-likelihood of an article's coordinates in the predicted distribution, which diverges wherever $q = 0$. we add a uniform background to the prediction, to ensure that it is non-zero everywhere:

$$\text{the predicted pdf is written} \qquad q' = \frac{1}{2}\left(\sum_{k=1}^{m} \frac{\mathbb{I}_k}{v_k} + q\right) \tag{11}$$

$$\text{the score for a study } \mathcal{S}_i = (\mathcal{T}_i, \mathcal{L}_i), \mathcal{L}_i = \{l_{i,a}\} \text{ is} \qquad \frac{1}{c_i} \sum_{a=1}^{c_i} \log(q'_i(l_{i,a})) \tag{12}$$

3 Empirical study

3.1 Data: mining neuroimaging publications

We downloaded roughly 140K neuroimaging articles from online sources including Pubmed Central and commercial publishers. About 14K of these contain coordinates, which we extracted, as in [4]. We built a vocabulary of around 1000 anatomical region names by grouping the labels of several atlases and the Wikipedia page "List of regions in the human brain"[3]. So in practice, $n \approx 14 \cdot 10^3$ and $d \approx 1000$. m depends on the atlas (or voxel grid) and ranges from 20 to 30K.

3.2 Text-to-brain encoding performance

Comparison of atlases and models. We perform 100 folds of shuffle-split cross-validation (10% in test set). As choices of $\{\mathcal{R}_k\}$, we compare several atlases

[3] https://en.wikipedia.org/wiki/List_of_regions_in_the_human_brain

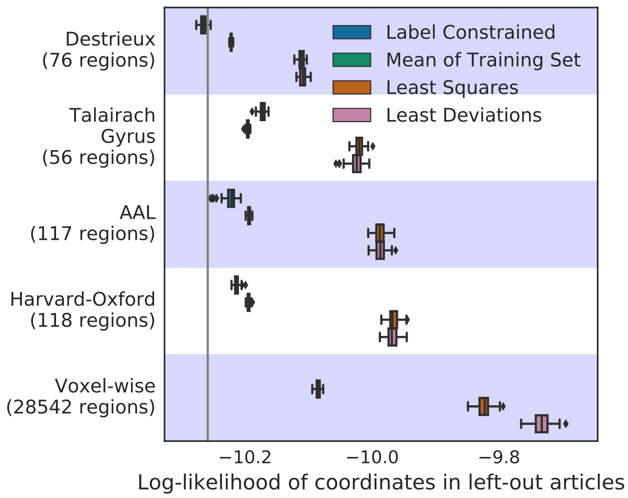


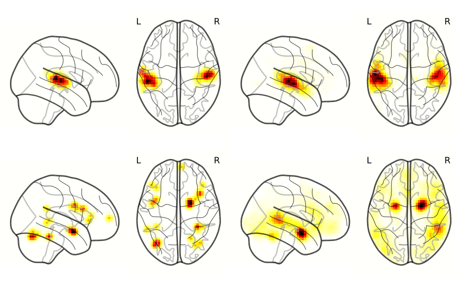
Fig. 1: **Log-Likelihood of coordinates reported by left-out articles in the predicted distribution (Eq. (12))**. The vertical line represents the test log-likelihood given a uniform distribution over the brain. Voxel-wise encoding is better than relying on any atlas. In this setting, ℓ_1 regression significantly outperforms least squares.

and a grid of cubic 4-mm voxels. We also compare ℓ_1 and ℓ_2 regression, and label-constrained ℓ_2. The label-constrained encoder is not used for the voxel grid, as it does not have labels. As a baseline, we include a prediction based on the average of the brain maps seen during training (i.e. independent of the text).

Fig. 1 gives the results: for all models, voxel-wise encoding performs better than any atlas. Large atlas regions regularize too much. Despite its higher dimensionality, voxel-wise encoding learns better representations of anatomical terms. The label-constrained model performs poorly, sometimes below chance, as the labels of a single atlas do not cover enough words and interactions between terms are important. For voxel-wise encoding, ℓ_1 regression outperforms ℓ_2. The best encoder is therefore learned using a ℓ_1 loss and a voxel partition.

Prediction examples. Fig. 2 shows the true pdf (estimated with KDE) and the prediction for the articles which obtained respectively the best and the first-quartile scores. The median is shown in the supplementary material.

Examples of coefficients learned by the linear regression. The coefficients of the linear regression (rows of $\boldsymbol{\beta}$) are the brain maps that the model associates



Best prediction: "Where sound position influences sound object representations: a 7-T fMRI study"

First quartile: "Interaction of catechol O-methyltransferase and serotonin transporter genes modulates effective connectivity in a facial emotion-processing circuitry."

Fig. 2: True map (left) and prediction (right) for best prediction and 1^{st} quartile

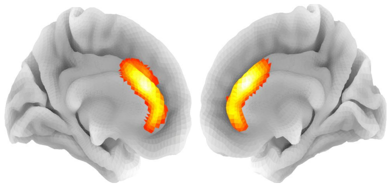


Fig. 3: regression coefficient for "anterior cingulate"

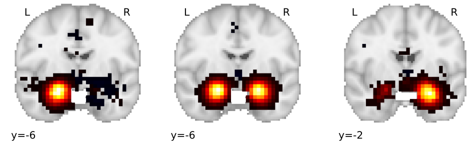


Fig. 4: regression coefficients for "left amygdala", "amygdala", and "right amygdala"

with each anatomical term. For frequent terms, they are close to what experts would expect (see for example Figs. 3 and 4).

3.3 Leveraging text without coordinates: neurological examples

Our framework can leverage unstructured spatial information contained in a large corpus of unannotated text. To showcase this, assume that we want to know which parts of the brain are associated with Huntington's disease. Our labelled corpus by itself is insufficient: only 21 documents mention the term "huntington". But we use it to learn associations between anatomical terms and locations in the brain (Section 2). This gives us access to the spatial information contained in the unlabelled corpus, which was out of reach before (Section 3.2). We contrast the mean encoding of articles which mention "huntington" against the mean distribution (taking the difference of their *log*). Since the large corpus contains more information about Huntington's disease (over 400 articles mention it), this is sufficient to see the striatum highlighted in the resulting map (Fig. 5, left). Fig. 5 (right) shows the experiment for Parkinson, and Fig. 6 for Aphasia.

4 Conclusion

We have introduced a theoretical framework to translate textual description of studies into spatial distributions over the brain. Such a translation enables pool-

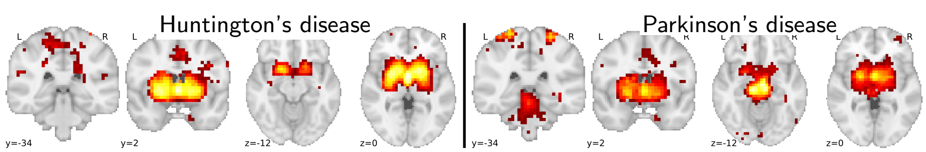


Fig. 5: **Predicted density for Huntington's and Parkinson's.** In agreement with Huntington's physiopathology [13], our method highlights the putamen, and the caudate nucleus. Also, in the case of Parkinson's [14], the brain stem, the thalamus, and the motor cortex are highlighted.

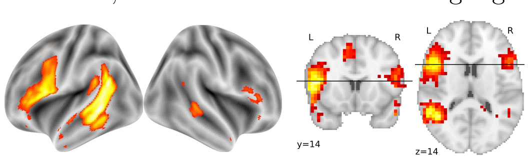


Fig. 6: **Predicted density for aphasia**, centered on Broca's and Wernicke's areas, in agreement with the literature [15].

ing together many studies which only provide text (no images or coordinates), for statistical analysis of their results in brain space. The statistical model gives a natural metric to validate. This metric enables comparing representations, showing that voxel-wise encoding is a better approach than relying on atlases. Building prediction models tailored to our task leads to a linear regression with an ℓ_1 loss (least absolute deviation), the total-variation distance between the true and the predicted spatial distributions. Such a model can be trained efficiently on dozens of thousands of data points and outperforms simpler approaches.

Applied to descriptions of pathologies that lack spatial information, our model synthesizes accurate brain maps that reflect the domain knowledge. Predicting spatial distributions of medical observations from text opens new alleys for clinical research from patient health records and case reports.

Acknowledgements This project received funding from: the European Union's H2020 Research Programme under Grant Agreement No. 785907 (HBP SGA2), the Metacog Digiteo project, the MetaMRI associate team, and ERC NeuroLang.

References

1. Mummery, C.J., Patterson, K., Price, et al.: A voxel-based morphometry study of semantic dementia: relationship between temporal lobe atrophy and semantic memory. Annals of neurology **47**(1) (2000) 36–45
2. Bohland, J., Bokil, H., Allen, C., Mitra, P.: The brain atlas concordance problem: quantitative comparison of anatomical parcellations. PloS one **4**(9) (2009) e7200
3. Laird, A.R., Fox, P.M., Price, C.J., et al.: ALE meta-analysis: Controlling the false discovery rate and performing statistical contrasts. Hum brain map **25** (2005) 155
4. Yarkoni, T., Poldrack, R.A., Nichols, T.E., et al.: Large-scale automated synthesis of human functional neuroimaging data. Nature methods **8** (2011) 665
5. Van der Zwaag, W., Gentile, G., et al.: Where sound position influences sound object representations: a 7-t fmri study. Neuroimage **54**(3) (2011) 1803–1811
6. Gibbs, A.L., Su, F.E.: On choosing and bounding probability metrics. International statistical review **70**(3) (2002) 419–435
7. Koenker, R., Bassett Jr, G.: Regression quantiles. Econometrica: journal of the Econometric Society (1978) 33–50
8. Chen, C., Wei, Y.: Computational issues for quantile regression. Sankhyā: The Indian Journal of Statistics (2005) 399–417
9. Koenker, R., d'Orey, V.: Remark as r92: A remark on algorithm as 229: Computing dual regression quantiles and regression rank scores. J Roy Stat Soc Series C **43**(2) (1994) 410–414
10. Portnoy, S., Koenker, R., et al.: The gaussian hare and the laplacian tortoise: computability of squared-error versus absolute-error estimators. Statistical Science **12**(4) (1997) 279–300
11. Yi, C., Huang, J.: Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. J Comp Graph Stat **26** (2017) 547
12. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J on Sci Comp **16**(5) (1995) 1190–1208
13. Walker, F.O.: Huntington's disease. The Lancet **369**(9557) (2007) 218–228
14. Davie, C.A.: A review of parkinson's disease. Br med bul **86** (2008) 109
15. Damasio, A.R.: Aphasia. New England Journal of Medicine **326**(8) (1992) 531–539

Supplementary material

A Factorization of the loss function

$$\begin{aligned}
\int_{\mathbb{R}^3} \delta(\hat{p}(z) - q(z))dz &= \int_{\mathbb{R}^3} \delta(\hat{p}(z) - q(z)) \sum_{k=0}^{m} \mathbb{I}_k(z)dz && (13)\\
&= \int_{\mathbb{R}^3} \delta(\hat{p}(z) - q(z)) \sum_{k=1}^{m} \mathbb{I}_k(z)dz \quad {}^{4} && (14)\\
&= \sum_{k=1}^{m} \int_{\mathbb{R}^3} \delta(\hat{\boldsymbol{y}}_k r_k(z) - \sum_{t=1}^{d} \boldsymbol{x}_t \boldsymbol{\beta}_{t,k} r_k(z))\mathbb{I}_k(z)dz \quad {}^{5} && (15)\\
&= \sum_{k=1}^{m} \int_{\mathbb{R}^3} \delta(\frac{\hat{\boldsymbol{y}}_k}{v_k} - \frac{\sum_{t=1}^{d} \boldsymbol{x}_t \boldsymbol{\beta}_{t,k}}{v_k})\mathbb{I}_k(z)dz && (16)\\
&= \sum_{k=1}^{m} v_k \delta(\frac{\hat{\boldsymbol{y}}_k}{v_k} - \frac{\sum_{t=1}^{d} \boldsymbol{x}_t \boldsymbol{\beta}_{t,k}}{v_k}) && (17)
\end{aligned}$$

B Derivation of the dual of penalized least-deviations

We have the minimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\hat{\boldsymbol{Y}} - \boldsymbol{X}\boldsymbol{\beta}\|_1 + \lambda\|\boldsymbol{\beta}\|_2^2 \right) \tag{18}$$

where $\boldsymbol{X} = (\boldsymbol{x}_i) \in \mathbb{R}^{n\times d}$ and $\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{y}}_i) \in \mathbb{R}^{n\times m}$ and $\lambda \in \mathbb{R}_+$.
The problem is equivalent to:

$$\underset{\boldsymbol{Z},\boldsymbol{\beta}}{\operatorname{argmin}} \left(\|\boldsymbol{Z}\|_1 + \lambda\|\boldsymbol{\beta}\|_2^2 \right) \tag{19}$$

$$\text{s.t. } \hat{\boldsymbol{Y}} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z} = 0 \tag{20}$$

Introducing the dual variable $\boldsymbol{\nu} \in \mathbb{R}^{n\times m}$, the Lagrangian is:

$$L(\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{\nu}) = \|\boldsymbol{Z}\|_1 + \lambda\|\boldsymbol{\beta}\|_2^2 + Tr(\boldsymbol{\nu}^T(\hat{\boldsymbol{Y}} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{Z})) \tag{21}$$

The derivative with respect to $\boldsymbol{\beta}$ is

$$2\lambda\boldsymbol{\beta} - \boldsymbol{X}^T\boldsymbol{\nu} \tag{22}$$

So minimizing with respect to $\boldsymbol{\beta}$ yields $\boldsymbol{\beta} = \frac{\boldsymbol{X}^T\boldsymbol{\nu}}{2\lambda}$ and

$$\min_{\boldsymbol{\beta}} L(\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{\nu}) = \|\boldsymbol{Z}\|_1 + \operatorname{Tr}(\boldsymbol{\nu}^T\hat{\boldsymbol{Y}} - \boldsymbol{\nu}^T\boldsymbol{Z} - \frac{1}{4\lambda}\boldsymbol{\nu}^T X X^T \boldsymbol{\nu}) \tag{23}$$

[4] because p and q are null in the background and $\delta(0, 0) = 0$
[5] because $\mathbb{I}_k \neq 0 \implies r_{k'} = 0 \ \ \forall k' \neq k$

The dual norm of the l_1 norm is l_∞, so minimizing with respect to $\boldsymbol{Z}$ we get the Lagrange dual function

$$g(\boldsymbol{\nu}) = \min_{\boldsymbol{Z},\boldsymbol{\beta}} L(\boldsymbol{Z}, \boldsymbol{\beta}, \boldsymbol{\nu}) = \begin{cases} Tr(\boldsymbol{\nu}^T \hat{\boldsymbol{Y}} - \frac{1}{4\lambda}\boldsymbol{\nu}^T XX^T \boldsymbol{\nu}) & \text{if } \|\boldsymbol{\nu}\|_\infty \leq 1 \\ -\infty & \text{otherwise} \end{cases} \tag{24}$$

The dual problem is:

$$\hat{\boldsymbol{\nu}} = \operatorname*{argmax}_{\boldsymbol{\nu}} \left(Tr(\boldsymbol{\nu}^T \hat{\boldsymbol{Y}} - \frac{1}{4\lambda}\boldsymbol{\nu}^T XX^T \boldsymbol{\nu}) \right) \quad \text{s.t. } \|\boldsymbol{\nu}\|_\infty \leq 1 \tag{25}$$

g is differentiable; its gradient is

$$\nabla_g(\boldsymbol{\nu}) = \hat{\boldsymbol{Y}} - \frac{1}{2\lambda}\boldsymbol{X}\boldsymbol{X}^T \boldsymbol{\nu} \tag{26}$$

And we solve this problem using an efficient algorithm: L-BFGS. Then we get back the primal solution as $\hat{\boldsymbol{\beta}} = \frac{\boldsymbol{X}^T \hat{\boldsymbol{\nu}}}{2\lambda}$. In practice, since data must be centered and normalized, the mean and scale of $\boldsymbol{X}$ appear in these formulas so that we do not break sparsity of $\boldsymbol{X}$: g is written

$$Tr(\boldsymbol{\nu}^T \tilde{\boldsymbol{Y}} - \frac{1}{4\lambda}\boldsymbol{K}^T \boldsymbol{K}) \tag{27}$$

With

$$\boldsymbol{K} = (\tilde{\boldsymbol{X}} - \bar{\boldsymbol{x}})^T \boldsymbol{\nu} = \tilde{\boldsymbol{X}}^T \boldsymbol{\nu} - \bar{\boldsymbol{x}} \odot \sum_{i=1}^{n} \boldsymbol{\nu}_i \tag{28}$$

Where $\tilde{\boldsymbol{X}} \in \mathbb{R}^{n \times d}$ is $\boldsymbol{X}$ divided by n times the variance of its columns, $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is the mean of the columns of $\boldsymbol{X}$ divided by the same quantity, $\tilde{\boldsymbol{Y}}$ is the centered and normalized $\hat{\boldsymbol{Y}}$, and $\odot$ is the Hadamard product. This is fast to compute because $\tilde{\boldsymbol{X}}$ is sparse (in practice, over 97% of entries are null). In a similar way, the gradient becomes

$$-\tilde{\boldsymbol{Y}} + \frac{1}{2\lambda}(\tilde{\boldsymbol{X}}\boldsymbol{K} - \bar{\boldsymbol{x}}\boldsymbol{K}) \tag{29}$$

and $\boldsymbol{\beta}$ is given by

$$\tilde{\boldsymbol{X}}^T \boldsymbol{\nu} - \bar{\boldsymbol{x}} \odot \sum_{i=1}^{n} \boldsymbol{\nu}_i \tag{30}$$

C More extensive atlas comparison

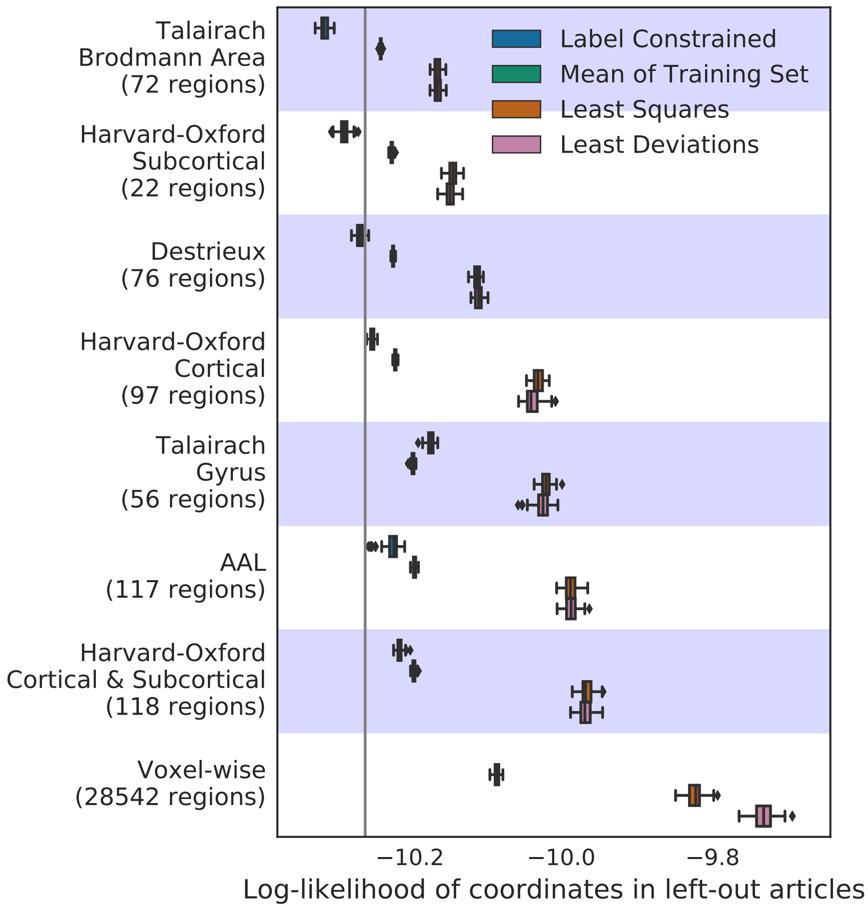
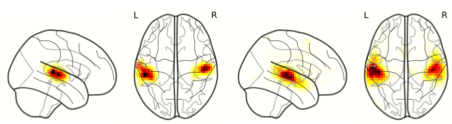


Fig. 7: Log-Likelihood of coordinates reported by left-out articles in the predicted distribution.

D Example predictions

D.1 Best prediction

Title: "Where sound position influences sound object representations: a 7-T fMRI study" [5]
Abstract: "Evidence from human and non-human primate studies supports a dual-pathway model of audition, with partially segregated cortical networks for sound recognition and sound localisation, referred to as the What and Where processing streams. In normal subjects, these two networks overlap partially on the supra-temporal plane, suggesting that some early-stage auditory areas are involved in processing of either auditory feature alone or of both. Using high-resolution 7-T fMRI we have investigated the influence of positional information on sound object representations by comparing activation patterns to environmental sounds lateralised to the right or left ear. While unilaterally presented sounds induced bilateral activation, small clusters in specific non-primary auditory areas were significantly more activated by contra-laterally presented stimuli. Comparison of these data with histologically identified non-primary auditory areas suggests that the coding of sound objects within early-stage auditory areas lateral and posterior to primary auditory cortex AI is modulated by the position of the sound, while that within anterior areas is not."

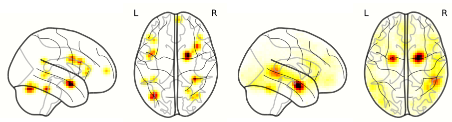


True map (left) and prediction (right)

D.2 First quantile

Title: "Interaction of catechol O-methyltransferase and serotonin transporter genes modulates effective connectivity in a facial emotion-processing circuitry." [17]
Abstract: "Imaging genetic studies showed exaggerated blood oxygenation level-dependent response in limbic structures in carriers of low activity alleles of serotonin transporter-linked promoter region (5-HTTLPR) as well as catechol O-methyltransferase (COMT) genes. This was suggested to underlie the vulnerability to mood disorders. To better understand the mechanisms of vulnerability, it is important to investigate the genetic modulation of frontal-limbic connectivity that underlies emotional regulation and control. In this study, we have examined the interaction of 5-HTTLPR and COMT genetic markers on effective connectivity within neural circuitry for emotional facial expressions. A total of 91 healthy Caucasian adults underwent functional magnetic resonance imaging experiments with a task presenting dynamic emotional facial expressions of fear, sadness, happiness and anger. The effective connectivity within the facial processing circuitry was assessed with Granger causality method. We have demonstrated that in fear processing condition, an interaction between 5-HTTLPR (S) and COMT (met) low activity alleles was associated with reduced

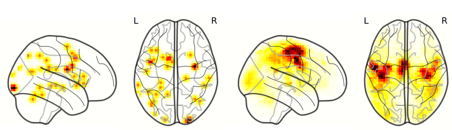
reciprocal connectivity within the circuitry including bilateral fusiform/inferior occipital regions, right superior temporal gyrus/superior temporal sulcus, bilateral inferior/middle prefrontal cortex and right amygdala. We suggest that the epistatic effect of reduced effective connectivity may underlie an inefficient emotion regulation that places these individuals at greater risk for depressive disorders."



True map (left) and prediction (right)

D.3 Median

Title: "How specifically are action verbs represented in the neural motor system: an fMRI study."[18]
Abstract: "Embodied accounts of language processing suggest that sensorimotor areas, generally dedicated to perception and action, are also involved in the processing and representation of word meaning. Support for such accounts comes from studies showing that language about actions selectively modulates the execution of congruent and incongruent motor responses (e.g., Glenberg & Kaschak, 2002), and from functional neuroimaging studies showing that understanding action-related language recruits sensorimotor brain areas (e.g. Hauk, Johnsrude, & Pulvermueller, 2004). In the current experiment we explored the basis of the neural motor system's involvement in representing words denoting actions. Specifically, we investigated whether the motor system's involvement is modulated by the specificity of the kinematics associated with a word. Previous research in the visual domain indicates that words denoting basic level category members lacking a specific form (e.g., bird) are less richly encoded within visual areas than words denoting subordinate level members (e.g., pelican), for which the visual form is better specified (Gauthier, Anderson, Tarr, Skudlarski, & Gore, 1997). In the present study we extend these findings to the motor domain. Modulation of the BOLD response elicited by verbs denoting a general motor program (e.g., to clean) was compared to modulation elicited by verbs denoting a more specific motor program (e.g., to wipe). Conform with our hypothesis, a region within the bilateral inferior parietal lobule, typically serving the representation of action plans and goals, was sensitive to the specificity of motor programs associated with the action verbs. These findings contribute to the growing body of research on embodied language representations by showing that the concreteness of an action-semantic feature is reflected in the neural response to action verbs."



True map (left) and prediction (right)

E Fast Kernel Density Estimation with convolutions

Kernel Density Estimation is a non-parametric way to estimate the pdf of a random variable, given a sample (possibly weighted). In our case the sample is the list of coordinates provided by a study. We use KDE to draw the brain map associated with each study. A reference on density estimation can be found in [19] or [20]. Once we have chosen a kernel ϕ (a function that sums to 1, symmetric and non-negative), we define the *rescaled kernel*:

$$\phi_{\boldsymbol{H}}(\boldsymbol{u}) = |\boldsymbol{H}|^{-\frac{1}{2}} \phi\left(\boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{u}\right) \tag{31}$$

where $|\cdot|$ is the determinant, and the smoothing parameter $\boldsymbol{H} \in \mathbb{R}^{d \times d}$ is a $(d \times d)$ symmetric positive definite matrix, called the *bandwidth matrix*.

The estimate of the pdf $\hat{p}$ is then given by:

$$\hat{p}(\boldsymbol{v}) = \frac{1}{c} \sum_{a=1}^{c} \omega_a \phi_{\boldsymbol{H}} (\boldsymbol{v} - \boldsymbol{l}_a) \tag{32}$$

where $\{\boldsymbol{l}_a, a = 1 \dots c\}$ is the sample and $\{\omega_a, a = 1 \dots c\}$ are the weights associated with the $\boldsymbol{l}_a$ and sum to c.

Note that if ϕ is the standard multivariate density (which is the kernel we use in practice), $\phi_{\boldsymbol{H}}$ is the multivariate density of mean 0 and covariance $\boldsymbol{H}$:

$$\phi(\boldsymbol{u}) = \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{1}{2} \boldsymbol{u}^T \boldsymbol{u}\right) \tag{33}$$

and

$$\begin{aligned}
\phi_{\boldsymbol{H}}(\boldsymbol{u}) &= |\boldsymbol{H}|^{-\frac{1}{2}} \phi\left(\boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{u}\right) && (34)\\
&= \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{H}|}} \exp\left(-\frac{1}{2} \boldsymbol{u}^T (\boldsymbol{H}^{-\frac{1}{2}})^T \boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{u}\right) && (35)\\
&= \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{H}|}} \exp\left(-\frac{1}{2} \boldsymbol{u}^T \boldsymbol{H}^{-1} \boldsymbol{u}\right) && (36)
\end{aligned}$$

In practice, we use a multivariate Gaussian kernel and isotropic smoothing (which is usually the case in neuroimaging):

$$\boldsymbol{H} = \begin{bmatrix} h^2 & & \\ & h^2 & \\ & & h^2 \end{bmatrix} \tag{37}$$

So Equation (32) can be rewritten as

$$\hat{p}(\boldsymbol{v}) = \frac{1}{ch^d} \sum_{a=1}^{c} \omega_a \phi\left(\frac{\boldsymbol{v} - \boldsymbol{l}_a}{h}\right) \tag{38}$$

with $d = 3$ is the dimension of the vector space in which our samples live.

A naive computation of this sum would be expensive. But since the estimate is the convolution of the kernel with the data, it can be computed efficiently with Fourier transforms [21,19]. In the general KDE setting, the sample points lie in a continuous space which needs to be binned in order to compute the discrete convolution. The grid we choose for binning will define the voxels of the brain map we compute. Two strategies are popular for computing the binned approximation of a weighted sample: simple binning and linear binning. Simple binning consists in assigning the whole weight of each sample point to the closest grid node. Linear binning distributes the weight of the sample point over the neighbouring nodes, e.g., in the one-dimensional case, if $\boldsymbol{v}$ lies between $i\delta$ and $(i+1)\delta$, (δ being the grid increment) the fraction of $\boldsymbol{v}$'s weight that $i\delta$ will receive is $\frac{(i+1)\delta-\boldsymbol{v}}{\delta}$. We simply assign the weights of the activation peaks to the voxel in which they lie. In other words, if an activation peak lies between $i\delta$ and $(i+1)\delta$, its whole weight will go to $i\delta$. Thus we compute the weights $w_{i,j,k}$ associated with the nodes of our grid (the voxels of our image) as

$$w_{i,j,k} = \sum_{a \in \mathcal{N}(i,j,k)} \omega_a \tag{39}$$

where ω_a is the weight associated with sample point l_a and where a belongs to the neighbourhood $\mathcal{N}(i,j,k)$ if and only if, writing $l_a = (x,y,z)$ the coordinates in $\mathbb{R}^3$ of l_a,

$$\begin{aligned} i\delta_x \leq x < (i+1)\delta_x \\ j\delta_y \leq y < (j+1)\delta_y \\ k\delta_z \leq z < (k+1)\delta_z \end{aligned}$$

where δ_x, δ_y and δ_z are the side lengths of the voxels.

The $w_{i,j,k}$ are sometimes called the *bin counts*. Equation (32) can be rewritten in voxel space as:

$$\hat{p}((i',j',k')) = \frac{1}{c} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{k=1}^{n_z} w_{i,j,k} \phi_{\boldsymbol{H}} \left(|i'-i|\delta_x, |j'-j|\delta_y, |k'-k|\delta_z \right) \tag{40}$$

where n_x, n_y and n_z are the dimensions of the image. This is the convolution of two images, one, $\boldsymbol{K}$, containing kernel evaluations at multiples of the voxel dimensions, and one, $\boldsymbol{W}$ containing the bin counts.

In [21], the authors present a scheme for zero-padding the matrices to convolve in a way that ensures very fast computation of their Fourier Transforms in the one-dimensional and two-dimensional cases; it generalizes immediatly to higher dimensions and we have used it for our three-dimensional images. This scheme is only valid when the bandwidth matrix $\boldsymbol{H}$ is diagonal; [22] recently described another padding which is adapted for unconstrained bandwidth matrices. However, the smoothing matrix we use is indeed diagonal and the procedure described in [21] can be applied in our case. We use the notation $\operatorname{diag}(\boldsymbol{H}) = (h_x, h_y, h_z)$

We use a Gaussian kernel, which has infinite support. However, it decreases very rapidly and we can approximate it by the truncated kernel, taking $\phi_{\boldsymbol{H}}(i\delta_x, j\delta_y, k\delta_z)$ to be 0 when $|\frac{i\delta_x}{h_x}| \geq 5$, $|\frac{j\delta_y}{h_y}| \geq 5$, and $|\frac{k\delta_z}{h_z}| \geq 5$ ([21] suggest 4 as a "safe choice"). This means that many of the terms in $\boldsymbol{K}$ can be set to zero.

Next, we define

$$\lambda_x = \min(n_x - 1, \left\lfloor \frac{5h_x}{\delta_x} \right\rfloor) \tag{41}$$

$$\lambda_y = \min(n_y - 1, \left\lfloor \frac{5h_y}{\delta_y} \right\rfloor) \tag{42}$$

$$\lambda_z = \min(n_z - 1, \left\lfloor \frac{5h_z}{\delta_z} \right\rfloor) \tag{43}$$

λ_x, λ_y and λ_z are the indices beyond which the kernel becomes practically null (or the dimensions of the image if these indices are superior to the image size). We also define

$$\theta_x = 2^{\lceil \log_2(n_x + \lambda_x + 1) \rceil} \tag{44}$$

$$\theta_y = 2^{\lceil \log_2(n_y + \lambda_y + 1) \rceil} \tag{45}$$

$$\theta_z = 2^{\lceil \log_2(n_z + \lambda_z + 1) \rceil} \tag{46}$$

θ_x is the smaller power of 2 that is bigger than $n_x + \lambda_x + 1$
Then the non-zero terms of $\boldsymbol{K}$ are symmetrized, and $\boldsymbol{K}$ is padded with zeros in the middle to size $(\theta_x \times \theta_y \times \theta_z)$. The reason for this padding is that the Fast Fourier Transform (FFT) is faster if the dimensions of $\boldsymbol{K}$ are highly composite numbers, such as powers of 2. The new matrix $\boldsymbol{K}$ is therefore a matrix of size $(\theta_x \times \theta_y \times \theta_z)$ such that:

$$\begin{aligned}
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}(i\delta_x, j\delta_y, k\delta_z), && i = 0, \ldots, \lambda_x, j = 0, \ldots, \lambda_y, k = 0, \ldots, \lambda_z \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}((\theta_x - i)\delta_x, j\delta_y, k\delta_z), && i = \theta_x - \lambda_x, \ldots, \theta_x - 1, j = 0, \ldots, \lambda_y, k = 0, \ldots, \lambda_z \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}(i\delta_x, (\theta_y - j)\delta_y, k\delta_z), && i = 0, \ldots, \lambda_x, j = \theta_y - \lambda_y, \ldots, \theta_y - 1, k = 0, \ldots, \lambda_z \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}(i\delta_x, j\delta_y, (\theta_z - k)\delta_z), && i = 0, \ldots, \lambda_x, , j = 0, \ldots, \lambda_y, k = \theta_z - \lambda_z, \ldots, \theta_z - 1 \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}(i\delta_x, (\theta_y - j)\delta_y, (\theta_z - k)\delta_z), && i = 0, \ldots, \lambda_x, j = \theta_y - \lambda_y, \ldots, \theta_y - 1, \\
& && k = \theta_z - \lambda_z, \ldots, \theta_z - 1 \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}((\theta_x - i)\delta_x, j\delta_y, (\theta_z - k)\delta_z), && i = \theta_x - \lambda_x, \ldots, \theta_x - 1, j = 0, \ldots, \lambda_y, \\
& && k = \theta_z - \lambda_z, \ldots, \theta_z - 1 \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}((\theta_x - i)\delta_x, (\theta_y - j)\delta_y, k\delta_z), && i = \theta_x - \lambda_x, \ldots, \theta_x - 1, j = \theta_y - \lambda_y, \ldots, \theta_y - 1, \\
& && k = 0, \ldots, \lambda_z \\
\boldsymbol{K}_{i,j,k} &= \phi_{\boldsymbol{H}}((\theta_x - i)\delta_x, (\theta_y - j)\delta_y, (\theta_z - k)\delta_z), && i = \theta_x - \lambda_x, \ldots, \theta_x - 1, \\
& && j = \theta_y - \lambda_y, \ldots, \theta_y - 1, \\
& && k = \theta_z - \lambda_z, \ldots, \theta_z - 1
\end{aligned}$$

For example, the first slice (:,:,0) of $\boldsymbol{K}$ looks like this:

$$
\boldsymbol{K}_{:,:,0} = \begin{bmatrix}
\phi_{0,0,0} & \phi_{0,1,0} & \dots & \phi_{0,\lambda_y,0} & 0 \dots 0 & \phi_{0,\lambda_y,0} & \dots & \phi_{0,1,0} \\
\phi_{1,0,0} & \phi_{1,1,0} & \dots & \phi_{1,\lambda_y,0} & 0 \dots 0 & \phi_{1,\lambda_y,0} & \dots & \phi_{1,1,0} \\
\vdots & \vdots & & \vdots & \vdots \quad \vdots & \vdots & & \vdots \\
\phi_{\lambda_x,0,0} & \phi_{\lambda_x,1,0} & \dots & \phi_{\lambda_x,\lambda_y,0} & 0 \dots 0 & \phi_{\lambda_x,\lambda_y,0} & \dots & \phi_{\lambda_x,1,0} \\
0 & 0 & \dots & 0 & 0 \dots 0 & 0 & \dots & 0 \\
\vdots & \vdots & & \vdots & \vdots \quad \vdots & \vdots & & \vdots \\
0 & 0 & \dots & 0 & 0 \dots 0 & 0 & \dots & 0 \\
\phi_{\lambda_x,0,0} & \phi_{\lambda_x,1,0} & \dots & \phi_{\lambda_x,\lambda_y,0} & 0 \dots 0 & \phi_{\lambda_x,\lambda_y,0} & \dots & \phi_{\lambda_x,1,0} \\
\vdots & \vdots & & \vdots & \vdots \quad \vdots & \vdots & & \vdots \\
\phi_{1,0,0} & \phi_{1,1,0} & \dots & \phi_{1,\lambda_y,0} & 0 \dots 0 & \phi_{1,\lambda_y,0} & \dots & \phi_{1,1,0}
\end{bmatrix} \tag{47}
$$

Where we have written $\phi_{i,j,k} = \phi_{\boldsymbol{H}}(i\delta_x, j\delta_y, k\delta_z)$ for brevity. The weights' matrix $\boldsymbol{W}$ is then padded with zeros to be the same size as $\boldsymbol{K}$: $\boldsymbol{W}$ is of size $(\theta_x, \theta_y, \theta_z)$ and

$$
\boldsymbol{W}_{i,j,k} = \begin{cases} w_{i,j,k} & \text{if } i < n_x, j < n_y, k < n_z \\ 0 & \text{otherwise} \end{cases} \tag{48}
$$

Then the upper corner of the convolution of $\boldsymbol{K}$ and $\boldsymbol{W}$ gives us the estimates of our pdf at every voxel:

$$
\hat{p}(I) = (\boldsymbol{K} * \boldsymbol{W})[: n_x, : n_y, : n_z] \tag{49}
$$

where $*$ is the convolution operator and I is the image:

$$
I = \{(i\delta_x, j\delta_y, k\delta_z), 0 \leq i < n_x, 0 \leq j < n_y, 0 \leq k < n_z\} \tag{50}
$$

Since the dimensions of $\boldsymbol{K}$ and $\boldsymbol{W}$ are highly composite numbers, computing their Fourier Transforms is very fast, and the convolution is obtained as the inverse Fourier Transforms of the element-by-element product of the transforms of $\boldsymbol{K}$ and $\boldsymbol{W}$, according to the discrete convolution theorem:

$$
\hat{f}(\text{image}) = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{K}) \odot \mathcal{F}(\boldsymbol{W})) \tag{51}
$$

Where $\mathcal{F}$ is the Fourier transform and $\odot$ is the Hadamard (element-wise) product.

When using KDE, two important decisions are the choice of the kernel and the choice of the smoothing factor h, also called bandwidth. Theoretical results as well as cross-validation can be used to make these choices [19]. We chose a Gaussian kernel for simplicity and because it is popular for smoothing MRI images. If we choose a bandwidth matrix with diagonal (h^2, h^2, h^2), the Full Width at Half Maximum (FWHM) of the kernel is equal to $2\delta h\sqrt{2\ln(2)}$, where δ is the voxel size. We compared $h = 0.5$, $h = 1$ and $h = 2$, and the bandwidth we chose is $h = 1$, which yields a FWHM of around 9 mm.

References

16. Van der Zwaag, W., Gentile, G., et al.: Where sound position influences sound object representations: a 7-t fmri study. Neuroimage **54**(3) (2011) 1803–1811
17. Surguladze, S., Radua, J., El-Hage, W., Gohier, B., Sato, J., Kronhaus, D., Proitsi, P., Powell, J., Phillips, M.: Interaction of catechol o-methyltransferase and serotonin transporter genes modulates effective connectivity in a facial emotion-processing circuitry. Translational psychiatry **2**(1) (2012) e70
18. Van Dam, W.O., Rueschemeyer, S.A., Bekkering, H.: How specifically are action verbs represented in the neural motor system: an fmri study. Neuroimage **53**(4) (2010) 1318–1325
19. Silverman, B.W.: Density estimation for statistics and data analysis. Volume 26. CRC press (1986)
20. Simonoff, J.S.: Smoothing methods in statistics. Springer Science & Business Media (2012)
21. Wand, M.: Fast computation of multivariate kernel estimators. J Comp Graph Stat **3**(4) (1994) 433–445
22. Gramacki, A., Gramacki, J.: Fft-based fast computation of multivariate kernel density estimators with unconstrained bandwidth matrices. J Comp Graph Stat (2016)